Approximate Pricing in Networks: How to Boost the Betweenness and Revenue of a Node

Ruben Brokkelkamp
Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands
Ruben.Brokkelkamp@cwi.nl

Sven Polak
Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Netherlands
sven_polak@hotmail.com

Guido Schäfer
Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands
Vrije Universiteit Amsterdam, Netherlands
g.schaefer@cwi.nl

Yllka Velaj
ISI Foundation, Turin, Italy
yllka.velaj@isi.it

Abstract

We introduce and study two new pricing problems in networks: Suppose we are given a directed graph $G = (V, E)$ with non-negative edge costs $(c_e)_{e \in E}$, k commodities $(s_i, t_i, w_i)_{i \in [k]}$ and a designated node $u \in V$. Each commodity $i \in [k]$ is represented by a source-target pair $(s_i, t_i) \in V \times V$ and a demand $w_i > 0$, specifying that w_i units of flow are sent from s_i to t_i along shortest s_i, t_i-paths (with respect to $(c_e)_{e \in E}$). The demand of each commodity is split evenly over all shortest paths. Assume we can change the edge costs of some of the outgoing edges of u, while the costs of all other edges remain fixed; we also say that we *price* (or *tax*) the edges of u.

We study the problem of pricing the edges of u with respect to the following two natural objectives: (i) *max-flow*: maximize the total flow passing through u, and (ii) *max-revenue*: maximize the total revenue (flow times tax) through u. Both variants have various applications in practice. For example, the max flow objective is equivalent to maximizing the *betweenness centrality* of u, which is one of the most popular measures for the influence of a node in a (social) network. We prove that (except for some special cases) both problems are NP-hard and inapproximable in general and therefore resort to approximation algorithms. We derive approximation algorithms for both variants and show that the derived approximation guarantees are best possible.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Graph algorithms analysis; Theory of computation → Shortest paths; Theory of computation → Network flows

Keywords and phrases Network pricing, Stackelberg network pricing, betweenness centrality, revenue maximization

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2019.13

Funding Part of this research was funded by the NWO Gravitation Project NETWORKS, Grant Number 024.002.003.

1 Introduction

Background and motivation. Nowadays, complex networks are used to model many different real-world scenarios and the analysis of these networks has become an extremely active research area. One of the main issues in complex network analysis is to identify the most “important” nodes in a network. To this aim, researchers have defined several centrality

© Ruben Brokkelkamp, Sven Polak, Guido Schäfer, and Yllka Velaj;
licensed under Creative Commons License CC-BY

measures to capture different notions of importance. One of the most popular measures is *betweenness centrality*, which ranks the nodes according to their frequency of occurrence on shortest paths between all possible pairs of nodes.

In several scenarios, having a high centrality can have a positive impact on the node itself. For example, in the context of social networks, Mahmoody et al. [22] show experimentally that nodes with high betweenness are also nodes that are highly influential when spreading information to other nodes in a social network. Valente and Fujimoto [33] claim that users with a high betweenness centrality (also called "brokers" or "bridging individuals") "may be more effective at changing others, more open to change themselves, and intrinsically interesting to identify". Moreover, in the field of transportation network analysis Malighetti et al. [23] analyze a network of 57 European airports and find that the betweenness centrality seems to be positively correlated to the efficiency of an airport. Also, increasing the betweenness of an airport would mean more traffic flowing through it and possibly attracting more customers for its shops.

The betweenness centrality notion can also be used to investigate problems arising in economics. For example, in the Netherlands it is an active debate whether the country is a tax haven for multinational enterprises. News articles headlined *"The Netherlands is a tax haven for many multinationals"* [14], *"The Netherlands is an attractive tax country"* [1], *"Dutch masters of tax avoidance"* [32] seem to lend support to the claim that this is indeed the case. To further investigate this question, the CPB (Netherlands Bureau for Economic Policy Analysis) has recently conducted some network analysis to identify important countries (using betweenness centrality) in the international tax treaty network [35, 36]; see also [30]. Among others, they conclude that companies mainly use the Netherlands as an intermediary country to send money through on a route from one country to another one. In this sense, the Netherlands is not a *tax haven*, i.e., a destination country where the money is stored (like the Bahamas or Bermuda), but a *conduit country*, i.e., an intermediary country on a route via which companies send their money.

In light of the above insight, a natural question that arises is how a country could maximize the amount of money that is sent through it. As a result, this would attract more jobs in the financial sector, or incentivize foreign companies to establish their businesses in the country (if only in the form of a letterbox). Another conceivable objective of a country might be to maximize the total amount of taxes that it obtains from the money transfers through it. These two questions constitute the main motivation for the network pricing problems studied in this paper.

Our contributions. In this paper, we introduce and study the following *Network Pricing Problem* (**NPP**): We are given a directed graph $G = (V, E)$ with non-negative edge costs $(c_e)_{e \in E}$, k commodities $(s_i, t_i, w_i)_{i \in [k]}$, a designated node $u \in V$ and a number $\kappa \geq 1$. Each commodity $i \in [k]$ is represented by a source-target pair $(s_i, t_i) \in V \times V$ and a demand $w_i > 0$, specifying that w_i units of flow are sent from s_i to t_i along shortest s_i, t_i-paths (with respect to $(c_e)_{e \in E}$). The demand of each commodity i is split evenly over all shortest s_i, t_i-paths. Suppose we can change the costs of $\kappa \leq \Delta(u)$ outgoing edges of u, where $\Delta(u)$ is the outdegree of u, while the costs of all other edges remain fixed; we also say that we *price* (or *tax*) the edges of u. Our goal is to optimally price at most κ edges of u such that (i) the total flow passing through u is maximized (**FLOW-NPP**), or (ii) the total revenue (i.e., flow times tax) through u is maximized (**REV-NPP**).

As it turns out, the problems behave rather differently in terms of hardness and approximability, depending on the objective under consideration and the parameter κ. More specifically, our main findings in this paper are as follows:

1. We show that **FLOW-NPP** can be solved in polynomial time when a constant number of edges or almost all edges of u can be priced.
2. In contrast, we prove that **FLOW-NPP** is NP-hard and $(1 - 1/e)$-inapproximable (even for the special case of unit demands) if κ is part of the input. Further, we show that a natural greedy algorithm achieves an approximation guarantee of $(1 - 1/e)$ (which is best possible).
3. We show that **REV-NPP** can be solved in polynomial time when only one edge can be priced. On the other hand, **REV-NPP** becomes NP-hard and $(1 - 1/e)$-inapproximable if κ is part of the input. We also show that the greedy algorithm might perform arbitrarily bad in this case.
4. We prove that **REV-NPP** is highly (meaning $1/\Delta(u)^{\varepsilon}$) inapproximable if all outgoing edges of u can be priced. We therefore focus on special cases of this problem.
   - First, we show that the single-commodity case is polynomial time solvable. This result also constitutes an important building block for our *uniform pricing algorithms* (i.e., all edges are priced the same).
   - Then, we focus on the unit demand setting and derive a (tight) H_k-approximate uniform pricing algorithm. We complement this result by showing that this problem is $1/\log^{\varepsilon}(k)$-inapproximable.
5. Finally, we show that our uniform pricing algorithm extends to the general setting and provides a $\max\{1/k, 1/\Delta(u)\}$-approximation algorithm for **REV-NPP** (which is essentially best possible).

Our results for **FLOW-NPP** mostly follow by using standard arguments for submodular function maximization. In contrast, we need to establish several new ideas and exploit structural insights to derive our results for **REV-NPP** (which constitutes the main technical contributions of this work).

We conclude with some (preliminary) experimental findings on an international tax treaty network based on real data. Our experiments indicate that our uniform pricing algorithm computes tax rates that would significantly increase the current tax revenue of the Netherlands (by a factor 68) and is at least within 51% of the optimal revenue (which is much better than the worst-case approximation guarantee suggests).

Related work. The problem of increasing the centrality of a node in a network has been widely investigated for different centrality measures. For example, boosting the popularity of web pages by increasing their page rank has been studied intensively [2, 26] with a particular focus on "fooling" search engines (e.g., through link farming [37]). The problem has also been considered for other centrality measures such as closeness centrality [11, 12], betweenness centrality [4], coverage centrality [13], eccentricity [15, 29], average distance [24] and some measures related to the number of paths passing through a given node [20]. Below, we give a few representative references only; most of these works focus on edge additions to increase the centrality.

Meyerson and Tegiku [24] give a constant factor approximation algorithm for the problem of minimizing the average shortest-path distance between all pairs of nodes by adding shortcut edges. Several algorithms are proposed in [27, 28] and experimentally shown to perform well in practice. Bauer et al. [3] study the problem of minimizing the average number of hops in shortest paths. They prove that the problem cannot be approximated within a logarithmic factor and provide respective approximation algorithms. Bilò et al. [5] and Demaine and Zadimoghaddam [15] consider the problem of minimizing the diameter of a graph and provide constant factor approximation algorithms.

The problem of maximizing revenue by pricing the edges of a graph has been studied in several works. These problems are known under different names such as the *network (or highway) pricing problem* [21, 9], but also as *Stackelberg network pricing games* [31, 8].

Labbe et al. [21] use a bilevel optimization model for taxing a given subset of the edges in a network to maximize the revenue that the leader receives from the followers. Among other results, they prove that the problem is NP-hard for single-commodity instances, exploiting negative edge costs and lower bound restrictions on the taxes. In a subsequent work, Roch et al. [31] improve upon this result and show NP-hardness for non-negative edge costs and without lower bound restrictions. They also provide an approximation algorithm for the single-commodity case.

Briest et al. [8] consider the following Stackelberg setting: There are several buyers who are interested in buying certain (pre-determined) subgraphs of the network and a seller (network owner) who can price a given subset of the edges. Once the seller fixes the prices, the buyers purchase the cheapest subgraph they are interested in. The goal is to maximize the total revenue obtained from the buyers. The authors show that a uniform price for all edges guarantees the seller a revenue within logarithmic factor of the optimal revenue. A more specific problem was considered by Briest et al. [7], where each buyer i is interested in purchasing a subgraph that contains a shortest s_i, t_i-path. Other special cases were considered in [18, 17, 19].

In general, there is a vast literature on the problem of pricing multiple items so as to maximize the revenue obtained from (possibly budget-constrained) buyers. There is a close connection between our problem and the problem of determining *envy-free* prices [19], because envy-freeness naturally corresponds to choosing the cheapest available option. Especially, we exploit known hardness results for the special cases of the *unit-demand pricing problem* and the *single-minded pricing problem* (see [19, 6, 10]) to establish the inapproximability results of our (more restrictive) network pricing problem.

We emphasize that our problem differs from the ones mentioned above because (i) the seller corresponds to a given node u who can set the prices of its outgoing edges only, and (ii) the revenue that u obtains depends on the proportion of the demand of each commodity routed along shortest paths through u.

2 Preliminaries

We formally define the *Network Pricing Problems* considered in this paper: Suppose we are given a directed graph $G = (V, E)$ with non-negative edge costs $(c_e)_{e \in E}$, k commodities $(s_i, t_i, w_i)_{i \in [k]}$[1], and a designated node $u \in V$. Each commodity $i \in [k]$ is specified by a source-target pair $(s_i, t_i) \in V \times V$ with $s_i \neq t_i$ and a non-negative demand (or weight) $w_i > 0$. The interpretation here is that each commodity $i \in [k]$ sends a total of w_i units of flow from the source node s_i to the target node t_i. The demand w_i is split evenly along all (simple[2]) shortest s_i, t_i-paths with respect to the edge costs $(c_e)_{e \in E}$ (formal definitions are given below). We assume that for each commodity $i \in [k]$, $s_i, t_i \neq u$ and there is at least one s_i, t_i-path that passes through u. This assumption is without loss of generality as otherwise the commodity is irrelevant (as will become clear below) and can be removed.

[1] Given an integer $k \geq 1$, we define $[k] = \{1, \ldots, k\}$.

[2] Recall that a path is said to be *simple* if it does not contain any cycles. Throughout the paper, whenever we refer to a shortest path we implicitly mean a simple shortest path.

We introduce some more notation. Let n and m be the the number of nodes and edges of G, respectively. We use the standard notation $\delta^+(u)$ to refer to the set of all outgoing edges of u, i.e., $\delta^+(u) = \{(u,v) \in E\}$, and define $\Delta(u) = |\delta^+(u)|$ as the outdegree of u. Given a pair of nodes $(x,y) \in V \times V$ with $x \neq y$, we denote by $\pi(x,y)$ the number of shortest x,y-paths with respect to $(c_e)_{e\in E}$. Similarly, we use $d(x,y)$ to refer to the total cost of a shortest x,y-path; we also say that $d(x,y)$ is the *distance* between x and y. For a set $A \subseteq E$, we denote by $d_{E\setminus A}(x,y)$ the distance between x and y in the graph $G = (V, E \setminus A)$, where the edges in A are removed. Below, we often omit the explicit reference to the respective edge costs if they are clear from the context.

For ease of notation, for every commodity $i \in [k]$, we use $\pi^i = \pi(s_i,t_i)$ to refer to the number of shortest s_i,t_i-paths. Further, we define π^i_u as the number of shortest s_i,t_i-paths that pass through node $u \in V$, where $s_i,t_i \neq u$. Given an outgoing edge $e = (u,v) \in E$ of u, we denote by π^i_e the number of shortest s_i,t_i-paths that pass through e. Observe that $\pi^i_u = \sum_{e\in\delta^+(u)} \pi^i_e$.

We can now define the flow that passes through the outgoing edges of u: Recall that the demand w_i of each commodity $i \in [k]$ is assumed to be split evenly over all shortest s_i,t_i-paths. Formally, the flow f^i_e of an outgoing edge $e = (u,v)$ of commodity i is defined as $f^i_e = w_i \cdot \pi^i_e/\pi^i$. The total flow passing through node u with respect to commodity i is then

$$f^i_u = \sum_{e\in\delta^+(u)} f^i_e = \sum_{e\in\delta^+(u)} w_i \cdot \frac{\pi^i_e}{\pi^i} = w_i \cdot \frac{\pi^i_u}{\pi^i}.$$

Further, we define $f_e = \sum_{i\in[k]} f^i_e$ as the total flow on edge e. The *total flow of node* u is then defined as

$$f_u = \sum_{e\in\delta^+(u)} f_e = \sum_{i\in[k]} \sum_{e\in\delta^+(u)} f^i_e = \sum_{i\in[k]} w_i \cdot \frac{\pi^i_u}{\pi^i} = \sum_{i\in[k]} f^i_u.$$

Another notion that is of interest in this paper is the following one: The *total revenue of node* u is defined as

$$r_u = \sum_{e\in\delta^+(u)} f_e \cdot c_e = \sum_{i\in[k]} \sum_{e\in\delta^+(u)} f^i_e \cdot c_e = \sum_{i\in[k]} \sum_{e\in\delta^+(u)} w_i \cdot \frac{\pi^i_e}{\pi^i} \cdot c_e.$$

Suppose we can change the costs of $\kappa \in [\Delta(u)]$ outgoing edges of u. How would we set the edge costs such that the total flow (or revenue, respectively) of u is maximized? More precisely, our goal is to determine a set $S \subseteq \delta^+(u)$ with $|S| \leq \kappa$ and non-negative costs $\bar{c}_S = (\bar{c}_e)_{e\in S}$ for the edges in S such that f_u (or r_u, respectively) with respect to the combined edge costs $(\bar{c}_S, c_{-S})$ is maximized, where we use $c_{-S} = (c_e)_{e\in E\setminus S}$ to refer to the (original) costs of the edges in $E \setminus S$ that remain unchanged. For convenience, we write $\bar{c}_e = \bar{c}_{\{e\}}$, we also write p_S when we set the cost of all edges in S to $p \in \mathbb{R} \cup \{\infty\}$. We use $f_u(\bar{c}_S)$ and $r_u(\bar{c}_S)$ to refer to the total flow and revenue of u, respectively, with respect to $(\bar{c}_S, c_{-S})$.

This gives rise to the following two optimization problems:

NETWORK PRICING PROBLEM (NPP)

Given:	A directed graph $G = (V,E)$ with non-negative edge costs $(c_e)_{e\in E}$, k commodities $(s_i,t_i,w_i)_{i\in[k]}$, a designated node $u \in V$ and a number $\kappa \in [\Delta(u)]$.
Goal:	Determine a set $S \subseteq \delta^+(u)$ with $|S| \leq \kappa$ and edge costs $\bar{c}_S = (\bar{c}_e)_{e\in S}$ such that $f_u(\bar{c}_S)$ is maximized (**FLOW-NPP**), or $r_u(\bar{c}_S)$ is maximized (**REV-NPP**)

Note that if the commodities correspond to all possible node pairs of the graph (not involving u as a source or target node), then the flow through u is precisely the betweenness centrality of u (as introduced above). In particular, in this case **FLOW-NPP** can be interpreted as the problem of maximizing the betweenness centrality of u.

In our discussion below, we distinguish the following three cases:

(C1) $\kappa = 1$: We are allowed to change the cost of only one outgoing edge of u.
(C2) $1 < \kappa < \Delta(u)$: We are allowed to change the cost of κ outgoing edges of u.
(C3) $\kappa = \Delta(u)$: We are allowed to change the cost of all the outgoing edges of u.

We continue with some basic observations. A pathological case we want to avoid in **REV-NPP** is that we can charge arbitrarily high costs.

▶ **Assumption 2.1.** For every commodity $i \in [k]$ there is at least one s_i, t_i-path that does not pass through u.

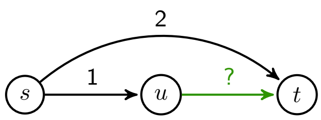


Figure 1 Example graph.

Throughout the paper, we assume that the edge costs are non-negative integers (as they may correspond to monetary units, percentages of a fixed precision, etc.).[3] The following example shows that this assumption is needed if one wants to be able to determine edge costs that realize the optimal revenue. Consider the instance depicted in Figure 1 and assume that there is a unit demand to be sent from s to t. Suppose we can impose an arbitrary non-negative rational cost $c_e \in \mathbb{Q}_{\geq 0}$ on the edge $e = (u, t)$. If we set $c_e = 1$, then the revenue of u becomes $\frac{1}{2}$. Otherwise, if we set $c_e = 1 - \varepsilon$ for a small rational $\varepsilon > 0$, then the revenue of u is $1 - \varepsilon$. It follows that **REV-NPP** does not admit an optimal solution.

Finally, we need to be able to efficiently compute how the flow splits. If there are zero cost cycles this may become infeasible [34]. We thus make the following assumption:

▶ **Assumption 2.2.** The edge costs $(c_e)_{e \in E}$ are non-negative integers and the graph does not contain any zero cost cycles, even if all outgoing edges of u are set to zero.

Using Assumption 2.2, it is possible to compute all relevant flows (as defined above) in polynomial time.[4] Throughout the paper we use this fact without stating it explicitly.

Due to space restrictions, some proofs are omitted from this extended abstract and can be found in the full version of the paper.

3 Flow Maximization Problem

In this section, we consider the problem **FLOW-NPP**. We first prove the following intuitive monotonicity property for the flow f_u through u: If the cost of a single outgoing edge of u decreases, then the flow through u does not decrease.

[3] All our results continue to hold if the edge costs are of the form $p \cdot \mathbb{Z}_{\geq 0}$ for some real number $p > 0$. In particular, this covers most practically relevant scenarios where one is bound to a finite number of decimals.

[4] This can be done by running for every commodity $i \in [k]$ an adapted version of Dijkstra's shortest path algorithm [16] which also counts the number of shortest paths passing through the edges.

▶ **Lemma 1.** *Consider an edge* $e = (u, v) \in \delta^+(u)$ *and assume that the edge cost* c_e *is decreased to* $\bar{c}_e < c_e$*. Then* $f_u(\bar{c}_e) \geq f_u(c_e)$*.*

Using Lemma 1, it is clear what we should do if we can price a subset $S \subseteq \delta^+(u)$ of edges: Simply set the cost of each edge $e \in S$ to zero to maximize the flow through u.

▶ **Corollary 2.** *Suppose we can change the costs of the edges in* $S \subseteq \delta^+(u)$*. Then setting* $\bar{c}_e = 0$ *for every* $e \in S$ *maximizes the flow* f_u *of* u*.*

Note that this takes away the difficulty of determining optimal costs for the edges in S. What remains is how to find the right subset of edges S to be priced. This is easy in cases **(C1)** and **(C3)**: It is not hard to see that by using complete enumeration over all possible subsets and Corollary 2, **FLOW-NPP** can be solved efficiently if $\kappa = \mathcal{O}(1)$ or $\kappa = \Delta(u) - \mathcal{O}(1)$.

▶ **Theorem 3.** **FLOW-NPP** *can be solved optimally in polynomial time for* $\kappa = \mathcal{O}(1)$ *and* $\kappa = \Delta(u) - \mathcal{O}(1)$*.*

We consider the cases of **(C2)** which are not captured by Theorem 3. Then the approach above fails. In fact, we show that **FLOW-NPP** is NP-hard to approximate within a factor $1 - 1/e$, even in the unit demand setting (i.e., $w_i = 1$ for all $i \in [k]$).

▶ **Theorem 4.** *Assuming* $P \neq NP$*, there is no* α*-approximation algorithm with* $\alpha > 1 - 1/e$ *for* **FLOW-NPP** *with* $\mathcal{O}(1) < \kappa < \Delta(u) - \mathcal{O}(1)$*, even in the unit demand setting.*

We derive a $(1 - 1/e)$-approximation algorithm for **FLOW-NPP**, which is best possible by Theorem 4. We use a well-known result due to Nemhauser et al. [25] for the following submodular function maximization problem: Given a finite set N, a function $z : 2^N \to \mathbb{R}$ and an integer k', find a set $S \subseteq N$ such that $|S| \leq k'$ and $z(S)$ is maximum. If z is non-negative, monotone and submodular[5], then the following natural greedy algorithm exhibits an approximation ratio of $1 - 1/e$ [25]: Start with the empty set and repeatedly add an element that gives the maximal *marginal gain*, i.e., if S is a partial solution, choose the element $j \in N \setminus S$ that maximizes $z(S \cup \{j\}) - z(S)$.

We show that $f_u(\bar{c}_S)$ (if considered as a set function) is non-negative, monotone and submodular.

▶ **Lemma 5.** *Define* $z(S) = f_u(0_S)$ *for every* $S \subseteq \delta^+(u)$*. The function* z *is non-negative, monotone and submodular.*

Algorithm 1 Greedy algorithm for **FLOW-NPP**.

```
S = ∅
for i = 1, ..., κ do
    e_max = arg max{f_u(0_{S∪{e}}) : e ∈ δ+(u) \ S}
    S = S ∪ {e_max}
return S
```

Applied to our setting, the greedy algorithm proceeds as described in Algorithm 1.

▶ **Theorem 6.** *The greedy algorithm provides a* $(1 - 1/e)$*-approximation for* **FLOW-NPP**.

[5] Let N be a finite set and let $z : 2^N \to \mathbb{R}$ be a function. Then z is (i) *non-negative* if $z(S) \geq 0$ for every $S \subseteq N$, (ii) *monotone* if $z(S) \leq z(T)$ for every $S \subseteq T \subseteq N$, and (iii) *submodular* if for all sets $S \subseteq T \subseteq N$ and every element $e \in N \setminus T$, $z(S \cup \{e\}) - z(S) \geq z(T \cup \{e\}) - z(T)$.

4 Revenue Maximization Problem

We turn to the problem **REV-NPP**. As it turns out, this problem is much more challenging than **FLOW-NPP**. In fact, even if we can change the costs of all outgoing edges of u it remains non-trivial to find good approximation algorithms (see Section 4.3).

4.1 Changing the cost of one edge

We consider case **(C1)** of **REV-NPP**, i.e., we can change the cost of one outgoing edge. We first show that we can efficiently compute the optimal cost if the edge is *given*.

▶ **Lemma 7.** *Fix an outgoing edge $e = (u, v)$ of u. We can then determine the cost $\bar{c}_e$ of e maximizing the revenue $r_u(\bar{c}_e)$ of u in polynomial time.*

Proof. Let $\bar{c}_e^*$ be some optimal cost which maximizes $r_u(\bar{c}_e^*)$. We first claim that there exists some optimal cost $\bar{c}_e$ with $\bar{c}_e \in T$, where

$$T = \left(\bigcup_{i\in[k]} \{T_i - 1, T_i\} \right) \cup \{\infty\} \quad \text{and} \quad T_i := d_{E\setminus\{e\}}(s_i, t_i) - d(s_i, u) - d(v, t_i) \ \ \forall i \in [k].$$

If $\bar{c}_e^* > \max\{T_i : i \in [k]\}$ there is no flow passing through e. We obtain the same by setting $\bar{c}_e = \infty \in T$ and thus $r_u(\bar{c}_e) = r_u(\bar{c}_e^*)$, which is optimal. Suppose now that $\bar{c}_e^* \leq \max\{T_i : i \in [k]\}$ and $\bar{c}_e^* \notin T$. Let $L = \{i \in [k] : T_i < \bar{c}_e^*\}$ and $U = \{i \in [k] : T_i - 1 > \bar{c}_e^*\}$. For the commodities in L there is no flow passing through e, while for the commodities in U the entire flow passes through e. By setting $\bar{c}_e = \min\{T_i - 1 : i \in U\}$ the flows do not change while $\bar{c}_e > \bar{c}_e^*$. Because $U \neq \emptyset$ we have $r_u(\bar{c}_e) > r_u(\bar{c}_e^*)$, contradicting the optimality of $\bar{c}_e^*$. Hence there is an optimal cost $\bar{c}_e$ in T.

Determining T takes at most $3k$ shortest path calculations. If all costs are fixed, we can compute the revenue by k shortest path calculations. Exploiting that $|T| \leq 2k + 1$, we can thus simply try all values in T and choose $\bar{c}_e$ as the cost that gives the largest revenue. ◀

By iterating over all edges $e = (u, v)$ of u and using Lemma 7 to determine the maximum revenue $r_u(\bar{c}_e)$, we can determine the optimal cost among all these edges. We obtain:

▶ **Theorem 8. REV-NPP(C1)** *can be solved optimally in polynomial time.*

4.2 Changing the costs of κ edges

We turn to case **(C2)** of **REV-NPP**. As we show, this problem is hard to approximate:

▶ **Theorem 9.** *Assuming $P \neq NP$, there is no α-approximation algorithm with $\alpha > 1 - 1/e$ for* **REV-NPP(C2)** *with $1 < \kappa < \Delta(u)$, even in the unit demand setting.*

One could hope that a greedy approach similar to the one used for **FLOW-NPP(C2)** would work here as well. Unfortunately, this is not the case. In fact, the greedy algorithm can perform arbitrarily bad. Further, the objective function is not submodular (even if the original costs are assumed to be $c_{\delta^+(u)} = \infty_{\delta^+(u)}$).

4.3 Changing the costs of all edges

We come to case **(C3)** of **REV-NPP**, where we are allowed to change the costs of all outgoing edges of u, i.e., $\kappa = \Delta(u)$. We start by proving some inapproximability results, both for the general and the unit demand setting, and then turn to our approximation algorithms.

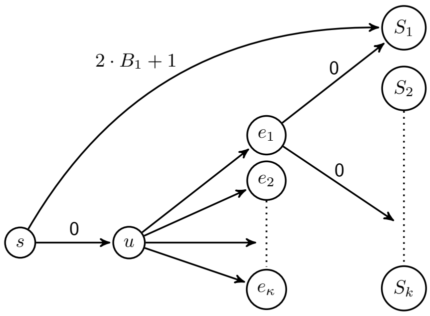


Figure 2 Illustration of the instance used in the proof of Theorem 10.

Inapproximability. Under reasonable hardness assumptions, case **REV-NPP(C3)** is hard to approximate within a factor of $\Omega(1/\log^\varepsilon(k))$ when considering unit demands and of $\Omega(1/\Delta(u)^\varepsilon)$ when considering arbitrary demands.

▶ **Theorem 10. REV-NPP(C3)** *is* $\Omega(1/\log^\varepsilon(k))$*-inapproximable for some* $\varepsilon > 0$ *in the unit demand setting, assuming that no polynomial-time algorithm can approximate constant-degree Balanced Bipartite Independent Set*[6] *to within arbitrarily small constant factors.* **REV-NPP(C3)** *is* $\Omega(1/\Delta(u)^\varepsilon)$*-inapproximable for some* $\varepsilon > 0$ *for arbitrary demands, assuming NP* $\subsetneq \cap_{\delta>0} BPTIME(2^{\mathcal{O}(n^\delta)})$.

We use a reduction from the *Unit-Demand Min-Buying Pricing Problem* (**UDP**$_{\min}$) [6]: We are given a set of N items $\mathcal{I} = \{e_1, \ldots, e_N\}$ and a set of k consumers $\mathcal{C} = \{c_1, \ldots, c_k\}$. Every consumer $c_i \in \mathcal{C}$ has some budget $B_i \in \mathbb{Z}_{\geq 0}$ and a set $S_i \subseteq \mathcal{I}$ of items she is interested in. Given prices $p : \mathcal{I} \to \mathbb{Z}_{\geq 0}$ for the items, consumer c_i will buy an item $e \in S_i$ with $p(e)$ minimum, but only if $p(e) \leq B_i$. The goal is to find prices that maximize the total revenue, i.e.,

$$\sum_{c_i \in \mathcal{C}} \min\{p(e) \,:\, e \in S_i \wedge p(e) \leq B_i\},$$

where we define the minimum of an empty set to be zero. In the so-called *economist's version* of **UDP**$_{\min}$ (**EUDP**$_{\min}$) we are additionally given a (discrete) probability distribution $\mathbb{P} : \mathcal{C} \to [0,1]$ over the consumers which is then incorporated in the objective function by multiplying the revenue gained from a consumer with her probability. Note that we can think of this probability distribution as having weights on the consumers.

Proof of Theorem 10. We give a reduction from **EUDP**$_{\min}$ to **REV-NPP** with arbitrary demands. The same reduction also provides the hardness result for uniform demands because we can see **UDP**$_{\min}$ as a special case of **EUDP**$_{\min}$, where all consumers have equal probabilities, and in what follows **UDP**$_{\min}$ is then reduced to **REV-NPP** with uniform demands.

We reduce an instance I of **EUDP**$_{\min}$ to an instance I' of **REV-NPP** such that any solution of I' can be converted into a solution of I losing at most a factor 2 in objective value. As a consequence, an α-approximation algorithm for **REV-NPP** with $\alpha = \Omega(1/\Delta(u)^\varepsilon)$ (respectively,

[6] In this problem, we are given a bipartite graph $G = (V, W, E)$ and we want to find maximum cardinality subsets of vertices $V' \subset V, W' \subset W$ with $|V'| = |W'|$, such that $\{v, w\} \notin E$ for all $v \in V'$, $w \in W'$; see [6] for more details.

$\alpha = \Omega(1/\log^{\varepsilon}(k)))$ provides an $\alpha/2$-approximation algorithm for **EUDP**$_{\min}$(respectively, **UDP**$_{\min}$). Briest [6] showed that the latter is not possible (under the assumptions stated in Theorem 10).[7]

Let $(\mathcal{I}, \mathcal{C}, (S_c)_{c\in\mathcal{C}}, (B_c)_{c\in\mathcal{C}}, \mathbb{P})$ be an instance of **EUDP**$_{\min}$. We construct an instance $I' = (G, (c_e)_{e\in E}, (s_i, t_i, w_i)_{i\in[k]}, u, \kappa)$ of **REV-NPP** as follows: Let the set of vertices of G be $V = \{s, u, S_1, \dots, S_k, e_1, \dots, e_N\}$, where each S_i, $i \in [k]$, and $e_j \in \mathcal{I}$ correspond to their counterpart in I. The set of edges E and the respective edge costs $(c_e)_{e\in E}$ are defined as follows (see Figure 2 for an illustration): There is an edge (s, u) of cost 0. For every S_j, $j \in [k]$, there is an edge which needs to be priced. For every $e_i \in \mathcal{I}$ and S_j, $j \in [k]$, such that $e_i \in S_j$ there is an edge (e_i, S_j) of cost 0. For every S_j, $j \in [k]$, there is an edge (s, S_j) of cost $2 \cdot B_j + 1$. Finally, we have k commodities (s, S_j, w_j) with demand $w_j = \mathbb{P}(c_j)$ for every $j \in [k]$. Note that $\Delta(u) = N$. Clearly, this reduction can be done in polynomial time.

First note that $\mathrm{OPT}(I') \geq 2\mathrm{OPT}(I)$ since taking the optimal prices p in I and using the prices $\bar{c}_{(u,e_i)} = 2p(e_i)$ for all $i \in [N]$ in I' will give a revenue of $2\mathrm{OPT}(I)$ for I'.

Consider a solution $\bar{c}$ for I' with some revenue Z'. We will convert this into a solution Z for I with value at least $Z'/4$. Note that we may assume that $B_j \geq 1$ for all $j \in [k]$ and therefore that $Z' \geq \sum_{i\in[k]} w_i \cdot 2 \min_{j\in[k]}\{B_j\} \geq 2\sum_{i\in[k]} w_i$ which is the revenue we would get by setting all prices to $2\min_{j\in[k]}\{B_j\}$. Now, modify $\bar{c}$ by subtracting 1 from $\bar{c}_e$ for all $e \in E$ if $\bar{c}_e$ is odd. This will cost us at most $\sum_{i\in[k]} w_i$ revenue. Thus we have $Z' - \sum_{i\in[k]} w_i \geq Z'/2$ revenue remaining. Observe that all prices are even and that $f_u^i/w_i \in \{0, 1\}$ for all $i \in [k]$. Using prices $p(e_i) = \bar{c}_{(u,e_i)}/2$ for $i \in [k]$ in I yields a revenue of at least $Z'/4$.

To conclude, if $Z' \geq \alpha\mathrm{OPT}(I')$ then $4Z \geq Z' \geq \alpha\mathrm{OPT}(I') \geq 2\alpha\mathrm{OPT}(I)$ implying $Z \geq \alpha/2\mathrm{OPT}(I)$ which proves the theorem. ◀

Special case: single commodity. We next consider the problem of **REV-NPP(C3)** for a single commodity only, i.e., $k = 1$. In this case, we can assume without loss of generality that $w_1 = 1$. Our goal is thus to determine $\bar{c}_{\delta^+(u)} = (\bar{c}_e)_{e\in\delta^+(u)}$ to maximize the revenue

$$r_u(\bar{c}_{\delta^+(u)}) = \sum_{e\in\delta^+(u)} f_e^1 \cdot \bar{c}_e = w_1 \sum_{e\in\delta^+(u)} \frac{\pi_e^1}{\pi_1} \cdot \bar{c}_e = \sum_{e\in\delta^+(u)} \frac{\pi_e^1}{\pi_1} \cdot \bar{c}_e.$$

▶ **Theorem 11. REV-NPP(C3)** *with a single commodity only (i.e., $k = 1$) can be solved optimally in polynomial time.*

Proof. For every edge $(u, v) \in \delta^+(u)$, we compute the value $h(v) := d_{E\setminus\delta^+(u)}(s_1, t_1) - d(s_1, u) - d(v, t_1)$. Let $T = \max_{(u,v)\in\delta^+(u)} h(v)$. If $T \leq 0$, then no revenue can be obtained and we stop. If $T > 0$, we compute the revenue obtained by setting all costs uniformly to either $T - 1$ or T:

$$\begin{aligned} r_u((T-1)_{\delta^+(u)}) &= T - 1 \\ r_u(T_{\delta^+(u)}) &= T \cdot \left(\textstyle\sum_{e=(u,v)\in\delta^+(u):h(v)=T} \pi_e^1\right)/\pi^1. \end{aligned}$$

[7] In fact, Briest [6] established the corresponding inapproximability results, where the prices and budgets are assumed to be reals. However, it is not hard to see that multiplying all budgets in the proof of Theorem 2 in [6] by a factor 2^k results in integer budgets. Then we can still assume that the prices are powers of 2, but now these powers are positive making also the prices integer. That is, the results in [6] also go through for integer values and budgets. We use this for our reduction here. Based on different assumptions, Chalermsook et al. [10] provide a stronger inapproximability result for the unit demand setting. If the same trick can be applied to the reduction presented in [10], our proof shows that the unit demand setting is $\log^{1-\epsilon}(k)$-inapproximable for every $\epsilon > 0$.

Algorithm 2 Uniform Price Algorithm.

$C = \emptyset$
foreach *commodity* $i \in [k]$ **do**
    Compute prices $T_i - 1$ and T_i as in Theorem 11 (considering only commodity i)
    Let $T_i^* \in \{T_i - 1, T_i\}$ be the price achieving higher revenue
    $C = C \cup \{T_i^*\}$
return $\arg\max\{r_u(p_{\delta^+(u)}) \,:\, p \in C\}$

We argue that the maximum of the two is the optimal revenue. When $\bar{c}_e > T$ for some $e \in \delta^+(u)$ it holds that $f_e = 0$, so we can assume that there is an optimum where $\bar{c}_e \leq T$ for all $e \in \delta^+(u)$. If there is an optimum for which $\bar{c}_e \leq T - 1$ for all $e \in \delta^+(u)$ then the maximum revenue we can get is $T - 1$ which is actually attained by setting all $\bar{c}_e$ to $T - 1$. Suppose the optimum is larger than $T - 1$ then there must be some $e' \in \delta^+(u)$ with $\bar{c}_{e'} = T$ and strictly positive flow. If there is some other edge for which $\bar{c}_e < T$ which gets flow then this contradicts $\bar{c}_{e'}$ getting flow, thus it cannot have flow in which case we could also set it to T. Thus then there must also be an optimum where $\bar{c}_e = T$ for all $e \in \delta^+(u)$. So, the maximum of $r_u(T_{\delta^+(u)})$ and $r_u((T-1)_{\delta^+(u)})$ is indeed the optimum. The values of $h(v)$ and $r_u(T_{\delta^+(u)})$ and $r_u((T-1)_{\delta^+(u)})$ can all be computed in polynomial time. ◀

Uniform pricing. We exploit the fact that for a single commodity we are able to find optimal uniform costs in polynomial time. Consider the *uniform price algorithm* described in Algorithm 2. First, we consider the case where all demands are uniform.

▶ **Theorem 12.** *Algorithm 2 is a $1/H_k$-approximation algorithm for* **REV-NPP(C3)** *when all demands are uniform and this is tight.*

Proof. We can assume without loss of generality that all demands are 1. Let T_i^* be the optimal price for commodity $i \in [k]$ as determined in the proof of Theorem 11 and let f_u^{i*} be the flow of commodity i going through u when using prices $\bar{c}_{\delta^+(u)} = (T_i^*)_{\delta^+(u)}$.

Assume that the commodities are ordered such that $T_1^* \geq T_2^* \geq \ldots \geq T_k^*$ and if $i < j$ and $T_i^* = T_j^*$ then $f_u^{i*} \geq f_u^{j*}$. So, first we order on T_i^* and if the T_i^* are equal then we order on f_u^{i*}. Let s be the number of unique values among the T_i^*. Let $i_1 = 1$ and define i_j for $2 \leq j \leq s$ recursively as the first entry that is strictly smaller than $T_{i_{j-1}}^*$. For convenience let $i_{s+1} = k + 1$.

Let P be the output of Algorithm 2. The algorithm tries prices T_i^* and because we have unit demands and by the ordering of the commodities we know that for $i \in \{i_j, \ldots, i_{j+1} - 1\}$, it holds that

$$P \geq T_i^* \cdot \left((i_j - 1) + \sum_{\ell = i_j}^{i_{j+1}-1} f_u^{\ell *} \right), \quad \text{which implies} \quad T_i^* \leq \frac{P}{(i_j - 1) + \sum_{\ell=i_j}^{i_{j+1}-1} f_u^{\ell *}}. \tag{1}$$

Let OPT be the maximum attainable revenue. If we single out the income from one commodity we cannot expect to do better than when we just consider that commodity. Hence,

$$\text{OPT} \le \sum_{i=1}^{k} T_i^* f_u^{i*} = \sum_{j=1}^{s} \sum_{i=i_j}^{i_{j+1}-1} T_i^* f_u^{i*} \le \sum_{j=1}^{s} \sum_{i=i_j}^{i_{j+1}-1} \frac{P \cdot f_u^{i*}}{(i_j - 1) + \sum_{\ell=i_j}^{i_{j+1}-1} f_u^{\ell*}} \tag{2}$$
$$\le \sum_{j=1}^{s} \sum_{i=i_j}^{i_{j+1}-1} \frac{P}{(i_j - 1) + (i - (i_j - 1))} = \sum_{j=1}^{s} \sum_{i=i_j}^{i_{j+1}-1} \frac{P}{i} = \sum_{i=1}^{k} \frac{P}{i} = H_k \cdot P$$

The second inequality follows from (1). For the third inequality we make use of the fact that $f_u^{i*} \le 1$ and that we sorted the commodities in such a way that if $i < j$ and $T_i^* = T_j^*$ we have $f_u^{i*} \ge f_u^{j*}$. Thus there are at least $i - (i_j - 1)$ terms for which the T^*-value is equal but the f-value is at least as large. ◀

We turn to the general demand case. We modify Algorithm 2 by replacing line 5 with $C = C \cup \{T_i - 1, T_i\}$.

▶ **Theorem 13.** *The modified version of Algorithm 2 is a* $\max\{1/k, 1/\Delta(u)\}$*-approximation algorithm for* **REV-NPP(C3)** *and this is tight.*

Proof. To show that it is a $1/k$-approximation algorithm we only need to consider the prices used in the original version of Algorithm 2. We follow the same reasoning as in Theorem 12. Let T_i^* and f_u^{i*}, $i \in [k]$, be as in the proof of Theorem 12. We note that $P \ge T_i^* \sum_{\ell=1}^{i} f_u^{\ell*}$, which implies that $T_i^* \le P / \sum_{\ell=1}^{i} f_u^{\ell*}$. Thus,

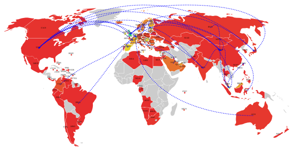
$$\text{OPT} \le \sum_{i=1}^{k} T_i^* \cdot f_u^{i*} \le \sum_{i=1}^{k} \frac{P \cdot f_u^{i*}}{\sum_{\ell=1}^{i} f_u^{\ell*}} \le \sum_{i=1}^{k} P = k \cdot P. \tag{3}$$

Next we show that the modified version of Algorithm 2 is also a $1/\Delta(u)$-approximation algorithm. Let $\bar{c}^*$ be the optimal prices giving a revenue of OPT. If we consider the revenue that is contributed by each $e \in \delta^+(u)$, there is at least one $e^* \in \delta^+(u)$ which contributed at least $\text{OPT}/\Delta(u)$, i.e., $f_{e^*} \cdot \bar{c}_{e^*}^* \ge \text{OPT}/\Delta(u)$. Consider $\bar{c}_e = \bar{c}_{e^*}^*$ for all $e \in \delta^+(u)$. The flow f_{e^*} will not go down because of $e \in \delta^+(u)$ such that $\bar{c}_e^* < \bar{c}_{e^*}^*$. Some of f_{e^*} may go to $e \in \delta^+(u)$ such that $\bar{c}_e^* \ge \bar{c}_{e^*}^*$ but if this happens we will still earn at least $\bar{c}_{e^*}^*$ on it. Hence the revenue for $\bar{c}_e$ is at least $f_{e^*} \cdot \bar{c}_{e^*}^* \ge \text{OPT}/\Delta(u)$.

Fix $\bar{c}_e = \bar{c}_{e^*}^*$ for all $e \in \delta^+(u)$. Let $\mathcal{F} = \{i \in [k] : f_u^i > 0\}$, i.e., all commodities that have some positive flow going through u and so we earn some revenue on them. Let T_i be the T corresponding to commodity i as in Theorem 11. Note that $\bar{c}_e^* \le \min\{T_i : i \in \mathcal{F}\}$. If $\bar{c}_e^* \ge \min\{T_i - 1 : i \in \mathcal{F}\}$ then we are done because then the approximation algorithm will try a price which yields at least $\text{OPT}/\Delta(u)$ revenue. Suppose $\bar{c}_e^* < \min\{T_i - 1 : i \in \mathcal{F}\}$. Then $f_u^i / w_i = 1$ for all $i \in \mathcal{F}$ and when raising $\bar{c}_e$ to $\min\{T_i - 1 : i \in \mathcal{F}\}$ for all $e \in \Delta(u)$ the flows for commodities $i \in \mathcal{F}$ will not change while the revenue increases. Hence the approximation algorithm tries a price which yields a revenue of at least $\text{OPT}/\Delta(u)$. We conclude that Algorithm 2 is a $1/\Delta(u)$-approximation algorithm. ◀

5 Conclusion

A motivating scenario for this research was figuring out how a country should change its tax rates in order to maximize its revenue. Computing the optimum is an intractable problem, but we can use our results to compute an optimal uniform tax. We used tax data from [35, 36], which also provides estimates of the volumes that are sent from one country to another (based



(a) Worldmap

	Revenue	
Original	0.0246140	} ×68
Uniform Pricing	1.6748451	
Optimum (UB)	3.2994390	} ×0.51
Optimal Uniform Tax	6.70%	

(b) Results

Figure 3 Outcome of experiments.

on the sizes of their economies). The data contains 108 countries (nodes), 8777 tax treaties (edges) and 11342 commodities. In this scenario, we need to find "money-transfer" paths such that the total tax paid by the companies is as low as possible. We run our experiments with "The Netherlands" as node u. The results are summarized in Figure 3. If the Netherlands would change its outgoing tax rate to 6.7% for all treaties, it would potentially increase its revenue by a factor 68. Further, the optimal uniform tax revenue is even within 51% of the optimum (upper bound as in (3)) and thus much better as suggested by Theorem 13.

We settle most cases of **FLOW-NPP** and **REV-NPP** in this paper but a case which is not completely settled is **REV-NPP(C2)**. Although we show that it is inapproximable within a factor $1 - 1/e$, case **(C3)** seems to suggest that it may even be harder.

An interesting way to look at our problem is from a game theory perspective. Now that we know what one node will do (approximately), what will happen if the nodes correspond to strategic players? Will they settle in a stable scenario where everybody gets some revenue, or will it end in a "price war" where the revenue of each player becomes zero?

References

1. Nederland is een aantrekkelijk belastingland. NOS, Nov. 6, 2014.
2. Konstantin Avrachenkov and Nelly Litvak. The Effect of New Links on Google Pagerank. *Stoc. Models*, 2006.
3. Reinhard Bauer, Gianlorenzo D'Angelo, Daniel Delling, Andrea Schumm, and Dorothea Wagner. The Shortcut Problem - Complexity and Algorithms. *J. Graph Algorithms Appl.*, 16(2):447–481, 2012.
4. Elisabetta Bergamini, Pierluigi Crescenzi, Gianlorenzo D'Angelo, Henning Meyerhenke, Lorenzo Severini, and Yllka Velaj. Improving the Betweenness Centrality of a Node by Adding Links. *ACM J. of Experimental Algorithmics*, 23, 2018.
5. Davide Bilò, Luciano Gualà, and Guido Proietti. Improved approximability and non-approximability results for graph diameter decreasing problems. *Theor. Comput. Sci.*, 417:12–22, 2012.
6. Patrick Briest. Uniform Budgets and the Envy-Free Pricing Problem. In *ICALP*, pages 808–819, 2008.
7. Patrick Briest, Parinya Chalermsook, Sanjeev Khanna, Bundit Laekhanukit, and Danupon Nanongkai. Improved Hardness of Approximation for Stackelberg Shortest-Path Pricing. In *Internet and Network Economics*, 2010.
8. Patrick Briest, Martin Hoefer, and Piotr Krysta. Stackelberg Network Pricing Games. *Algorithmica*, 62(3), 2012.
9. Luce Brotcorne, F. Cirinei, Patrice Marcotte, and Gilles Savard. An exact algorithm for the network pricing problem. *Discrete Optimization*, 8(2):246–258, 2011.
10. Parinya Chalermsook, Julia Chuzhoy, Sampath Kannan, and Sanjeev Khanna. Improved Hardness Results for Profit Maximization Pricing Problems with Unlimited Supply. In *APPROX-RANDOM*, pages 73–84, 2012.

11 Pierluigi Crescenzi, Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. Greedily improving our own centrality in a network. In *SEA 2015*, volume 9125 of *Lecture Notes in Computer Science*, pages 43–55, 2015.

12 Pierluigi Crescenzi, Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. Greedily Improving Our Own Closeness Centrality in a Network. *ACM Trans. Knowl. Discov. Data*, 11(1):9:1–9:32, 2016.

13 Gianlorenzo D'Angelo, Martin Olsen, and Lorenzo Severini. Coverage Centrality Maximization in Undirected Networks. *CoRR*, abs/1811.04331, 2019. `arXiv:1811.04331`.

14 P. de Waard. Nederland belastingparadijs voor veel multinationals. De Volkskrant, Oct. 14, 2011.

15 Erik D. Demaine and Morteza Zadimoghaddam. Minimizing the Diameter of a Network Using Shortcut Edges. In *SWAT*, volume 6139 of *Lecture Notes in Computer Science*, pages 420–431, 2010.

16 Edsger W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numer. Math.*, 1(1):269–271, 1959.

17 Iftah Gamzu and Danny Segev. A Sublogarithmic Approximation for Highway and Tollbooth Pricing. In *Proc. 37th International Colloquium Conference on Automata, Languages and Programming*, ICALP'10, pages 582–593, 2010.

18 Fabrizio Grandoni and Thomas Rothvoss. Pricing on Paths: A PTAS for the Highway Problem. *SIAM J. Comput.*, 45(2):216–231, 2016.

19 Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On Profit-maximizing Envy-free Pricing. In *SODA*, pages 1164–1173, 2005.

20 Vatche Ishakian, Dóra Erdös, Evimaria Terzi, and Azer Bestavros. A Framework for the Evaluation and Management of Network Centrality. In *Proc. 12th SIAM Int. Conf. on Data Mining (SDM)*, pages 427–438, 2012.

21 Martine Labbe, Patrice Marcotte, and Gilles Savard. A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing. *Management Science*, 44(12):1608–1622, December 1998.

22 Ahmad Mahmoody, Charalampos E. Tsourakakis, and Eli Upfal. Scalable Betweenness Centrality Maximization via Sampling. In *Proc. 22nd ACM SIGKDD Int. Conf. on KDD*, pages 1765–1773, 2016.

23 Paolo Malighetti, Gianmaria Martini, Stefano Paleari, and Renato Redondi. The Impacts of Airport Centrality in the EU Network and Inter-Airport Competition on Airport Efficiency. Technical Report MPRA-7673, University Library of Munich, Germany, 2009.

24 Adam Meyerson and Brian Tagiku. Minimizing Average Shortest Path Distances via Shortcut Edge Addition. In *APPROX*, volume 5687, 2009.

25 George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions–I. *Math. Program.*, 14(1):265–294, 1978.

26 Martin Olsen and Anastasios Viglas. On the approximability of the link building problem. *TCS*, 518:96–116, 2014.

27 Manos Papagelis, Francesco Bonchi, and Aristides Gionis. Suggesting Ghost Edges for a Smaller World. In *Proc. 20th ACM Int. Conf. on Information and Knowledge Management*, CIKM '11, pages 2305–2308. ACM, 2011.

28 Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. Selecting Shortcuts for a Smaller World. In *Proc. SIAM Int. Conf. on Data Mining*, pages 28–36. SIAM, 2015.

29 Senni Perumal, Prithwish Basu, and Ziyu Guan. Minimizing Eccentricity in Composite Networks via Constrained Edge Additions. In *Military Communications Conference, MILCOM 2013 - 2013 IEEE*, pages 1894–1899, 2013.

30 Sven C. Polak. Algorithms for the Network Analysis of Bilateral Tax Treaties. MSc. thesis, University of Amsterdam, The Netherlands, 2014.

31 Sébastien Roch, Gilles Savard, and Patrice Marcotte. An approximation algorithm for Stackelberg network pricing. *Networks*, 46(1):57–67, 2005.

32 D. Milmo S. Goodley. Dutch masters of tax avoidance. The Guardian, Oct. 19, 2011.

33 Thomas W. Valente and Kayo Fujimoto. Bridging: Locating critical connectors in a network. *Soc. Netw.*, 32(3):212–220, 2010.

34 Leslie Valiant. The Complexity of Enumeration and Reliability Problems. *SICOMP*, 8(3):410–421, 1979.

35 Maarten van 't Riet and Arjan M. Lejour. Ranking the Stars: Network Analysis of Bilateral Tax Treaties. CPB Discussion Paper, 2014.

36 Maarten van 't Riet and Arjan M. Lejour. Optimal tax routing: network analysis of FDI diversion. *International Tax and Public Finance*, 25(5):1321–1371, 2018.

37 Baoning Wu and Brian D. Davison. Identifying link farm spam pages. In *Proc. 14th Int. Conf. on World Wide Web, WWW 2005*, pages 820–829, 2005.